E-Hash: Herramienta de Software para Dispersión de Archivos.

Ariel Sobrado, Luciano Marrero, Rodolfo Bertone, Pablo Thomas
asobrado@gmail.com {lmarrero,pbertone,pthomas}@lidi.info.unlp.edu.ar

Instituto de Investigación en Informática LIDI
Facultad de Informática UNLP

Resumen: El avance de la tecnología informática, tanto en software como en hardware, ha logrado ubicar a los sistemas de Bases de Datos (BD) como el anfitrión necesario de cualquier organización al momento de persistir información digitalizada. El estudio y comprensión de una BD debe comenzar por comprender las estructuras de datos que le sirven de soporte. El dictado de la asignatura Introducción a las Bases de Datos (IBBDD) de la Facultad de Informática de la UNLP abarca diversos temas relacionados con organización de Archivos, entre ellos, Dispersión de Archivos. Si bien la asignatura brinda explicaciones y consultas de las guías prácticas, algunas veces éstas resultan insuficientes para madurar conceptos. Además, no se dispone de una herramienta de software que pueda asistir al alumno en la comprensión del tema Dispersión de Archivos. El propósito de E-HASH (Herramienta de Software para la enseñanza de técnicas de Dispersión de Archivos) es proporcionar al alumno de un tutor virtual en el aprendizaje del tema mencionado, con el marco conceptual establecido en IBBDD.

Keywords: Bases de Datos, Dispersión de archivos, Hashing, Herramienta Educativa.

1 Introducción

En el mundo de la informática conceptos tales como: velocidad de respuesta, eficiencia en la búsqueda de información, cantidad de accesos a memoria secundaria y otros conceptos, fueron y son motivos de estudio e investigación con el objetivo de obtener mejores prestaciones. Obviamente, las BD no quedan fuera de estos términos al momento de persistir, organizar y gestionar la información sobre almacenamiento secundario.

El acceso a la información en memoria secundaria es considerado un proceso extremadamente lento (en términos relativos) comparado con las altas velocidades de acceso a memoria principal o primaria (memoria RAM); por lo tanto, minimizar los accesos a memoria secundaria para encontrar un elemento implica menor tiempo de respuesta.

Existen tres modos principales de acceder a la información en un sistema de archivos que integra una BD: en forma secuencial, a través de un índice, y de manera directa.

En general, los costos adicionales en el mantenimiento de archivos al utilizar índices, pueden reducirse lo suficiente como para que sean aceptables en término de acceso a disco, pero hay ocasiones en que la demanda de un sistema de archivos es tan extrema que dichos costos se vuelven intolerables, es decir, resulta absolutamente necesaria una forma de acceso que requiera menos de dos accesos a disco (en promedio) por cada operación sobre un registro. En estos casos, el acceso directo por clave deberá ser la organización de archivos elegida, siendo la Dispersión (o Hashing) el método utilizado para tal fin [1].

La organización de Archivos por Dispersión permite encontrar cualquier elemento del Archivo con sólo un acceso a disco, salvo situaciones especiales. La mayoría de las operaciones se logran con un sólo acceso, aunque este ideal no siempre sea alcanzable.

La técnica de dispersión presenta dos características importantes en lo que respecta a una organización de archivos. Estas características son:

- No requiere almacenamiento de información adicional. Si se elige Dispersión como método de organización del archivo, es el mismo archivo de datos el que realmente resulta disperso [1], [2].
- Reduce al mínimo la cantidad de accesos a memoria secundaria en las operaciones de inserción, búsqueda y eliminación de elementos en el Archivo. En general, con un solo acceso a disco se puede almacenar, encontrar o eliminar un elemento del Archivo. Si bien no es posible asegurar que cualquier elemento sea almacenado o encontrado en un solo acceso, la gran mayoría de estas operaciones serán efectivamente resueltas respetando dicha premisa.

También es importante mencionar que la Dispersión de Archivos, al igual que cualquier otra organización de Archivos, presenta algunas limitaciones, entre ellas:

- No es posible aplicar la técnica de Dispersión en aquellos Archivos con registros de longitud variable, o al menos su implantación resulta en algoritmos complejos.
- No existe un orden lógico de los elementos dispersados.

2 Dispersión de Archivos

2.1 Marco teórico

Dispersión significa aplicar una función F(K), donde K es una cadena o valor numérico denominado clave (en este caso deberá ser la clave primaria) que permite identificar un único elemento del archivo de datos. Dicha función da como resultado una dirección que pertenece al rango de direcciones permitidas, es decir,

0<=*F(K)*<=M (donde M es el tamaño máximo del espacio de direcciones permitido). La dirección generada por la función F(K) se denomina dirección base de la clave K y se utiliza para la búsqueda, inserción y eliminación de registros en el Archivo.

Muchas veces al termino Dispersión se lo denomina "aleatorización" dado que no existe una relación obvia inmediata entre la clave que se está dispersando y la dirección resultante luego de haberle aplicado la función.

Existen dos tipos de Dispersión de Archivos de acuerdo a la configuración establecida para el espacio de direcciones. Estos dos tipos son: Dispersión con espacio de direccionamiento estático y Dispersión con espacio de direccionamiento dinámico. La diferencia radica en que en el primero el espacio disponible para dispersar los elementos del archivo de datos es fijado previamente por el usuario, así, la función de dispersión solo genera valores para este rango de direcciones. Con el segundo tipo de Dispersión el espacio de direcciones para dispersar los elementos del Archivo de datos crece o decrece de acuerdo a las necesidades de espacio que requiera dicho Archivo en cada momento determinado

2.2 Tratamiento de Colisiones

El primer problema que surge cuando se dispersa un Archivo son las claves sinónimas. Cuando F(k) = F(Y), donde K<>Y, se dice que K e Y son claves sinónimas para la función F.

Una colisión se produce cuando a dos o más claves distintas la función de dispersión les asigna la misma dirección en el Archivo de datos. Es decir, que dos o más elementos del Archivo intentan ser almacenados en la misma dirección.

Además, como cada dirección tiene una capacidad finita, si una dirección recibe una clave más de las que puede almacenar, esta última clave produce un problema denominado desborde, saturación u overflow.

Si se produce un desborde, la clave residirá fuera de su dirección original asignada, denominada dirección base, ocupando otra dirección del Archivo de datos. Si bien se debe pensar en cómo se resolverán los desbordes, antes se debe tratar de minimizar la cantidad de colisiones producidas. Para ello se debe elegir una función de Dispersión que distribuya las claves de manera lo más uniforme posible, aunque es imposible encontrar un algoritmo o función de Dispersión que genere una distribución uniforme.

Otra alternativa es la posibilidad de ampliar el espacio de memoria en disco, o permitir almacenar más de un registro por dirección o nodo. Obviamente, es importante analizar el costo de la configuración elegida.

2.3 Tratamiento de Desbordes

Cuando el número de colisiones iguala a la cantidad de registros que se pueden almacenar en una dirección, ocurre un problema denominado desborde, saturación u overflow. Esto significa que el registro a insertar en el Archivo no cabe en su dirección base y se debe decidir donde realmente se almacenará.

Cuando ocurre un desborde se debe realizar dos acciones: encontrar lugar para el registro a insertar, y asegurar que ante una búsqueda dicho registro será efectivamente encontrado. Para esto existen diversos métodos o técnicas

A continuación se explicarán brevemente algunas de las técnicas más difundidas.

Saturación Progresiva: este método pertenece al tipo de dispersión estática y es considerado, por su implementación, el método más sencillo para resolver un desborde. Consiste en almacenar el nuevo registro en la dirección siguiente más próxima de su dirección base. Esto significa que cuando sucede un desborde se examina secuencialmente las direcciones siguientes, partiendo de la dirección base correspondiente y hasta encontrar un lugar disponible. Cuando se busca un registro o un espacio libre y se llega al final del espacio de direcciones, se vuelve a iniciar desde la primera dirección de dicho espacio. La gran ventaja de la saturación progresiva es su simplicidad, pero si ocurren muchas colisiones habrá varios registros en saturación que ocupen espacios distintos a sus direcciones base e irán formando cúmulos de registros. Esto produce como consecuencia mayor cantidad de accesos a disco para extraerlos.

Saturación Progresiva Encadenada: este método es similar a la Saturación Progresiva y está diseñado para evitar el problema causado por el acumulamiento de registros, el cual afecta directamente a la búsqueda de registros en el Archivo. La diferencia con la Saturación Progresiva es que una vez localizada la nueva dirección para el registro en saturación, la misma se encadena o enlaza con la dirección base de dicho registro, generando así una cadena de registros que poseen claves sinónimas para la función de Dispersión utilizada.

El resultado final es que para cada conjunto de claves sinónimas hay una lista ligada, interna al Archivo de datos, que conecta sus registros, siendo en esta lista en la que se busca cuando se requiere encontrar un registro. Esto último es la ventaja principal del método con respecto a la Saturación Progresiva. Como contrapartida, se requiere un campo de liga a cada registro del Archivo de datos, por lo tanto, se requiere mayor espacio de almacenamiento.

Dispersión Doble: este método está diseñado para resolver el problema causado por el acumulamiento de registros. Cuando más "lleno" se encuentre el Archivo, más registros acumulados habrá en direcciones "vecinas", provocando búsquedas muy largas.

El método consiste en disponer de dos funciones de Dispersión. La primera función produce la dirección de base en la cual el registro será ubicado. En el caso que exista un desborde se utiliza la segunda función de Dispersión. Esta segunda función se aplica también a la clave, pero no retorna precisamente una dirección, sino que retorna un valor que representa un desplazamiento. Este desplazamiento se suma a la dirección base originada por la primera función, y esta suma dará como resultado una dirección en el espacio de memoria disponible, donde se intentará almacenar la clave en desborde. En caso de generarse nuevamente desborde, se deberá sumar reiteradamente el desplazamiento obtenido, y así sucesivamente hasta encontrar una dirección con espacio suficiente para albergar al registro.

Saturación Progresiva Encadenada con Área de Desborde Separada: en los métodos anteriores a medida que ocurren los desbordes los registros son reubicados en direcciones distintas a su dirección base, ocupando así, las potenciales direcciones base de otros registros. Este método sugiere como alternativa utilizar un área de memoria separada para aquellos registros que provoquen desborde. Es decir, hay dos tipos de direcciones, las que son direccionadas por la función de Dispersión y las que se reservan sólo para los registros que generen desborde. La ventaja de este método es que se mantienen libres para futuras altas las direcciones que son base. Generalmente, el conjunto de direcciones base es mucho mayor que el conjunto de direcciones destinadas a los desbordes; la razón de esto es que los desbordes son casos excepcionales.

2.4 Dispersión Extensible

A medida que la BD crece, los Archivos de gran volumen producen muchas colisiones y se genera una sobrecarga en el acceso a los registros. Una de las estrategias para mitigar este problema es reservar desde el comienzo un espacio estimado para requisitos futuros; con el potencial desperdicio de espacio de almacenamiento.

Otra solución consiste reorganizar el Archivo a medida que crezca, sin la reserva de espacio a priori. El método que se presenta para este tipo de organización de Archivos se denomina Dispersión Extensible.

La Dispersión Extensible presenta una alternativa de implementación con espacio de direccionamiento dinámico. El método consiste en utilizar espacio a medida que se lo necesite, es decir, que no existe un tamaño de memoria secundaria predefinida y reservada, sino que ésta crece o disminuye de acuerdo al Archivo de datos a dispersar. Este método utiliza una sola función de dispersión que retorna para cada clave una cadena de bits, la cual va a determinar donde se deberá almacenar el registro. Para la implementación del método es necesaria una estructura auxiliar administrada en memoria principal. Esta estructura contiene para una cierta cadena de bits, la dirección física en disco en la cual se almacenará aquella clave que por función de dispersión retorne dicha cadena de bits [3], [4].

3 Marco teórico del trabajo realizado

Este trabajo presenta el desarrollo de una herramienta de software educativa, la cual se denominó E-Hash (Herramienta de Software para la Enseñanza de Técnicas de Dispersión de Archivos), con el propósito de asistir a los alumnos en el aprendizaje de Dispersión de Archivos, con el marco conceptual impuesto por la asignatura Introducción a las Bases de Datos de la Facultad de Informática de la UNLP.

En la experiencia como docentes de la cátedra se ha observado que si bien se alcanza la comprensión general del tema mencionado, es muy conveniente disponer

de un software asistente que actúe como una herramienta complementaria en el proceso enseñanza-aprendizaje.

Actualmente, los alumnos resuelven las guías prácticas con el método tradicional de lápiz y papel para luego evacuar sus dudas con los auxiliares docentes, con quienes corroboran la precisión de la interpretación y resolución de sus ejercicios. La utilización de E-Hash no pretende reemplazar la práctica tradicional sino que tiene como propósito actuar como complemento, fortaleciendo la actividad de enseñanza y aprendizaje del tema tratado. Mediante E-Hash el alumno podrá analizar la resolución un problema, generando el caso de uso y comprobando su resolución paso a paso.

E-Hash es una herramienta de software portable, con una interface visual amigable, que el alumno podrá utilizar en el proceso de aprendizaje [5], [6].

4 E-Hash: Herramienta de Software para Dispersión de Archivos.

E-Hash es una herramienta Web educativa. Está desarrollada con software de uso libre, y pretende ser un producto que actúe como complemento para el proceso de enseñanza y aprendizaje del tema Dispersión de Archivos [5], [6].

La herramienta presenta una interface evolutiva e intuitiva organizada en solapas para que el alumno pueda avanzar paso a paso en la configuración del ambiente de Dispersión. La figura 1 muestra la interface inicial de E-hash.

En la solapa de inicio se presenta un breve texto y un menú de ayuda, el cual se mantiene siempre visible independientemente de la solapa que se encuentre activa.

El primer parámetro a configurar es el indicado en la, solapa “Función de Dispersión”, donde se presentan tres posibilidades: “modulo tamaño de memoria”, “centros cuadrados” y “transformación de la base”. Para esta última función el alumno puede seleccionar la base a la que desea convertir la clave.

Fig. 1: Interface inicial de E-Hash

En la solapa “Espacio de memoria” se deben configurar los parámetros referentes al espacio de memoria: cantidad de direcciones/nodos o densidad de empaquetamiento, capacidad de una dirección/nodo y cantidad de claves que se van a dispersar. La exclusión entre la cantidad de direcciones y la densidad de empaquetamiento está dada porque con una de ellas es posible calcular la otra. La figura 2 muestra la selección de los parámetros correspondientes al espacio de memoria.

Fig. 2: selección de parámetros correspondiente al espacio en memoria

Se debe definir, además, la técnica a utilizar en la resolución de colisiones con desborde dentro del ambiente de dispersión. E-Hash permite seleccionar cuatro técnicas con administración de memoria estática y una técnica con administración de memoria dinámica. “Estas son: Saturación Progresiva”, “Saturación Progresiva Encadenada”, ”Saturación Progresiva con Área de Desborde Separada”, “Dispersión Doble” y “Dispersión Extensible” respectivamente.

Fig. 3: Interface inicial de la simulación

Finalizada la configuración del ambiente de Dispersión se presenta la simulación de la operatoria sobre Archivos. El alumno puede realizar operaciones de inserción, eliminación y búsqueda de claves. La figura 3 presenta la interface inicial de simulación.

En esta figura, sobre el margen derecho se encuentra el panel de operaciones, el campo de texto en donde el alumno podrá ingresar las claves a dispersar, el historial de las operaciones realizadas y el clásico menú de ayuda. Sobre el margen izquierdo se puede observar los valores seleccionados para la configuración elegida del ambiente de Dispersión. En la parte central se grafica el espacio de memoria en disco, donde se muestra el estado del cómo va quedando el Archivo de datos dispersado, a medida que se realizan las operaciones correspondientes. Inicialmente no hay localidad de memoria alguna ocupada.

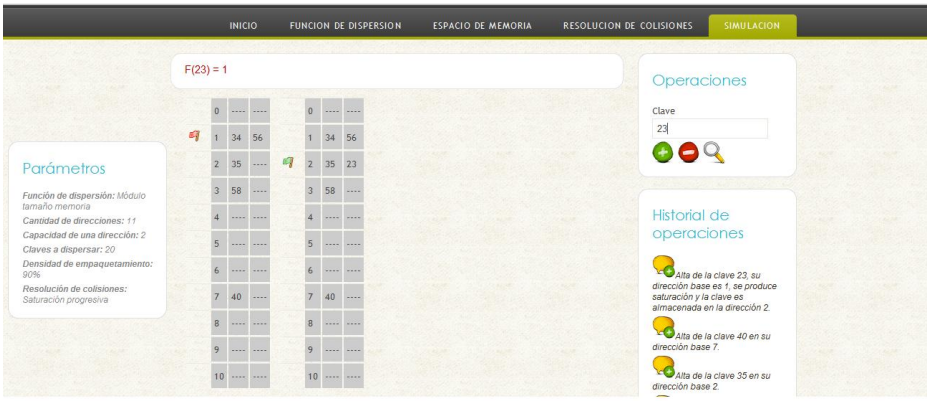

Fig. 4: Alta con desborde

Cuando se selecciona una de las tres posibles operaciones a realizar con la clave ingresada, se inicia la simulación resultado de aplicar la función de Dispersión. Para el caso de alta, se muestra cómo se dispersa la clave en su dirección base y si existiera un desborde, la simulación transcurre en base al método de resolución de colisiones con desborde seleccionado. La figura 4 muestra un alta con desborde respectivamente.

La interface de E-Hash fue creada considerando que se trata de una herramienta educativa. De esta forma, se logró un ambiente amigable, de fácil lectura y simple de utilizar. Para lograr una separación entre la lógica de programa y interface se utilizaron hojas de estilo en cascada CSS (Cascading Style Sheets).

5 Conclusiones

La Dispersión constituye una de las estrategias más importantes para organizar Archivos de datos. Con esta estrategia se logra una organización de Archivos con acceso directo. Esto se debe a que para la mayoría de las operaciones (alta, eliminación, modificación o consulta) se necesita en promedio menos de dos accesos a memoria secundaria.

Uno de los objetivos previstos en la asignatura Introducción a las Bases de Datos es que el alumno comprenda los beneficios de este tipo de organización de Archivos, junto a como se desarrollan los algoritmos para las operaciones ante mencionadas.

La construcción de E-Hash busca proveer de una herramienta interactiva que le permita al alumno resolver los ejercicios definidos en las guías prácticas, a fin de agilizar el proceso de aprendizaje. El alumno puede plantear sus propias configuraciones y simular la operatoria sobre archivos en un ambiente de animación adecuado.

La herramienta actualmente está en una etapa de prueba y ha sido el resultado de la experiencia de docentes de la asignatura, así como de las reuniones de brainstorming llevadas a cabo.

La evolución actual del producto permite suponer que durante el año 2012, E-Hash estará disponible para ser utilizado por los alumnos en la asignatura (estimativamente 600 alumnos), y por ende ser evaluada en profundidad.

6 Trabajos Futuros

En una primera etapa, y luego de una prueba de campo, se espera poder optimizar la interface y animación, de acuerdo a los requerimientos de usabilidad planteados por los alumnos.

Además, se prevé adicionar nuevas funciones de Dispersión y nuevas técnicas de resolución de colisiones con desborde.

7 Bibliografía

1. Michael Folk, Bill Zoellick, Greg Ricciardi. Estructuras de Archivos. Addision Wesley 1992. ISBN: 0-201-62923-2.
2. Rodolfo Bertone, Pablo Thomas. Introducción a las Bases de Datos. Fundamentos y Diseño. Pearson Latinoamérica 2011. ISBN: 978-987-615-136-8.
3. Peter Smith, Michael Barnes. Files & Databases: An Introduction. Addison Wesley 1987. ISBN: 0-201-10746-5.
4. Gary W. Hansen, James V. Hansen. Diseño y Administración de Bases de Datos. Prentice Hall 1997. ISBN: 84-8322-002-4.
5. Rodolfo Bertone, Emanuel Nucilli, Pablo Thomas. HEA: Herramienta de Software para enseñanza de árboles. XVI Congreso Argentino de Ciencias de la Computación. Universidad de Morón. Año 2010.
6. R. Bertone, P. Thomas, S. Antonetti, A. Miglio. Herramienta para la enseñanza de Modelado Conceptual de Bases de Datos. XV Congreso Argentino de Ciencias de la Computación. Universidad Nacional de Jujuy. Año 2009.